A proposal to upgrade the CRS batch system

Tomasz Wlodek

The current CRS system

- Written long time ago
- Works well
- It can serve RHIC for years to come
- But there are some problems:
- From time to time jobs disappear without trace

Current CRS system

- The source of the problems is believed to be an outdated Perl communications library
- It needs to be replaced by newer product
- This requires extensive rewriting
- So why not to rewrite all of CRS?

What could we gain by rewriting CRS?

- **New code will be in Python : easier to maintain** *(Yes, I know that Perl people will protest this statement…)*
- **We could add new functions, as requested by users**
- **We could try to make the new CRS GRID enabled…**

Possible scenarios

- **Minimal:** just rewrite CRS in Python, more or less as is.
- **Maximal:** Build a unified general system for everything which could serve 4 RHIC experiments and ATLAS and be practical realization of GRID
- **Realistic:** Something in between the first two

Before we go any further: Why do we need CRS at at all? Why not use LSF,PBS, etc?

- The answer is: HPSS. No batch system can pre-stage data from HPSS even before a job is submitted. CRS does that.
- The need to optimize the staging of data from HPSS in order to minimize the number of tape mounts makes the CRS the only option for RHIC experiments.

CRS and ATLAS

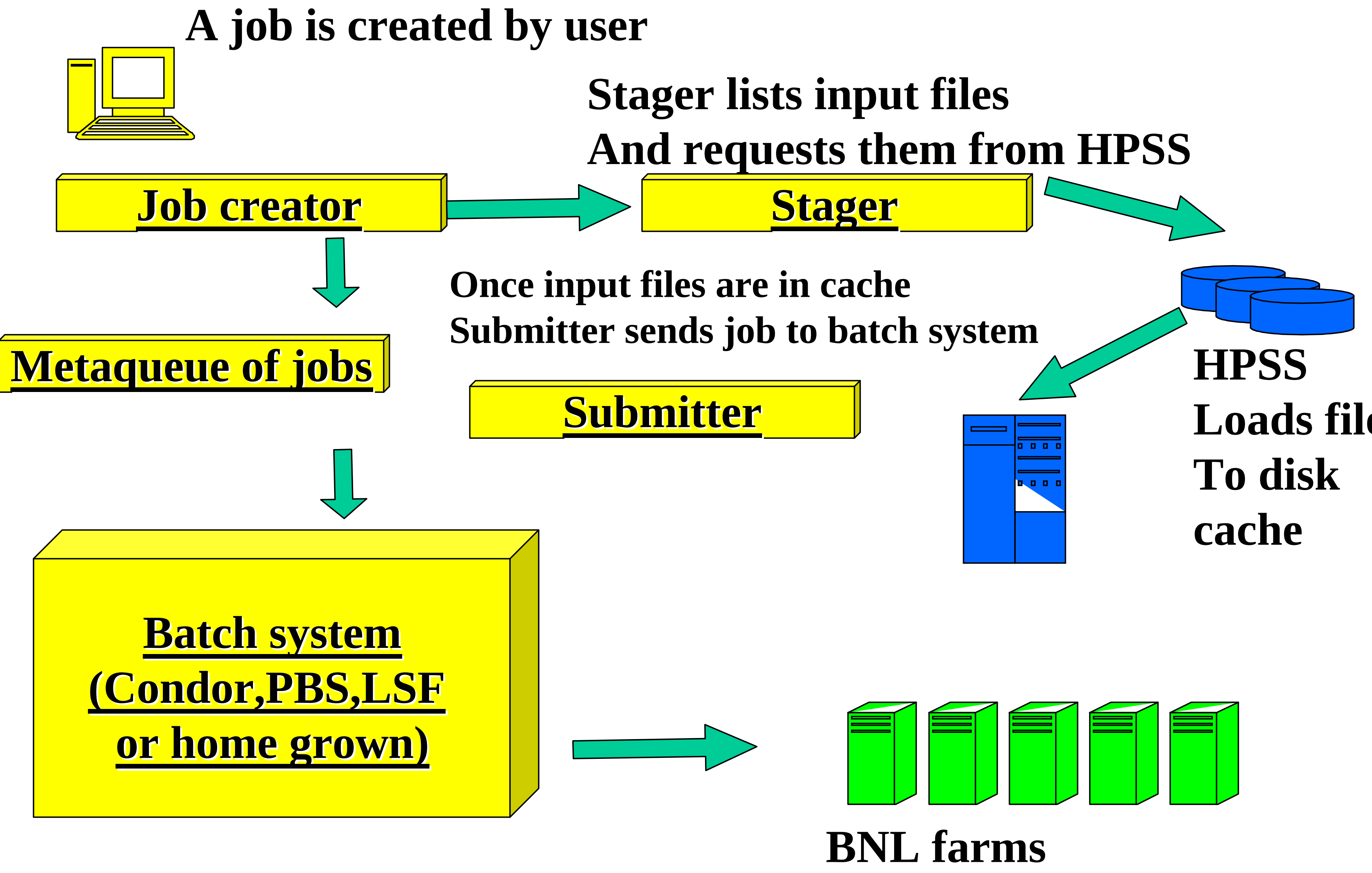
- At present Atlas experiment does not make great use of HPSS
- Atlas is not interested in specialized CRS batch software
- Atlas GRID testbed relies heavily on Globus technology, which is not present in CRS
- As a result the present CRS software has no use for Atlas
- Could we somehow make CRS relevant for Atlas experiment?

CRS serves dual purpose:

1. **It manages the data staging from and to HPSS storage**
2. **It runs the jobs on the farm and monitors them**

- **Those two functions can be separated: staging can be done by our own code, but running jobs can be delegated to and off-shelf batch system**

So, what do we propose?



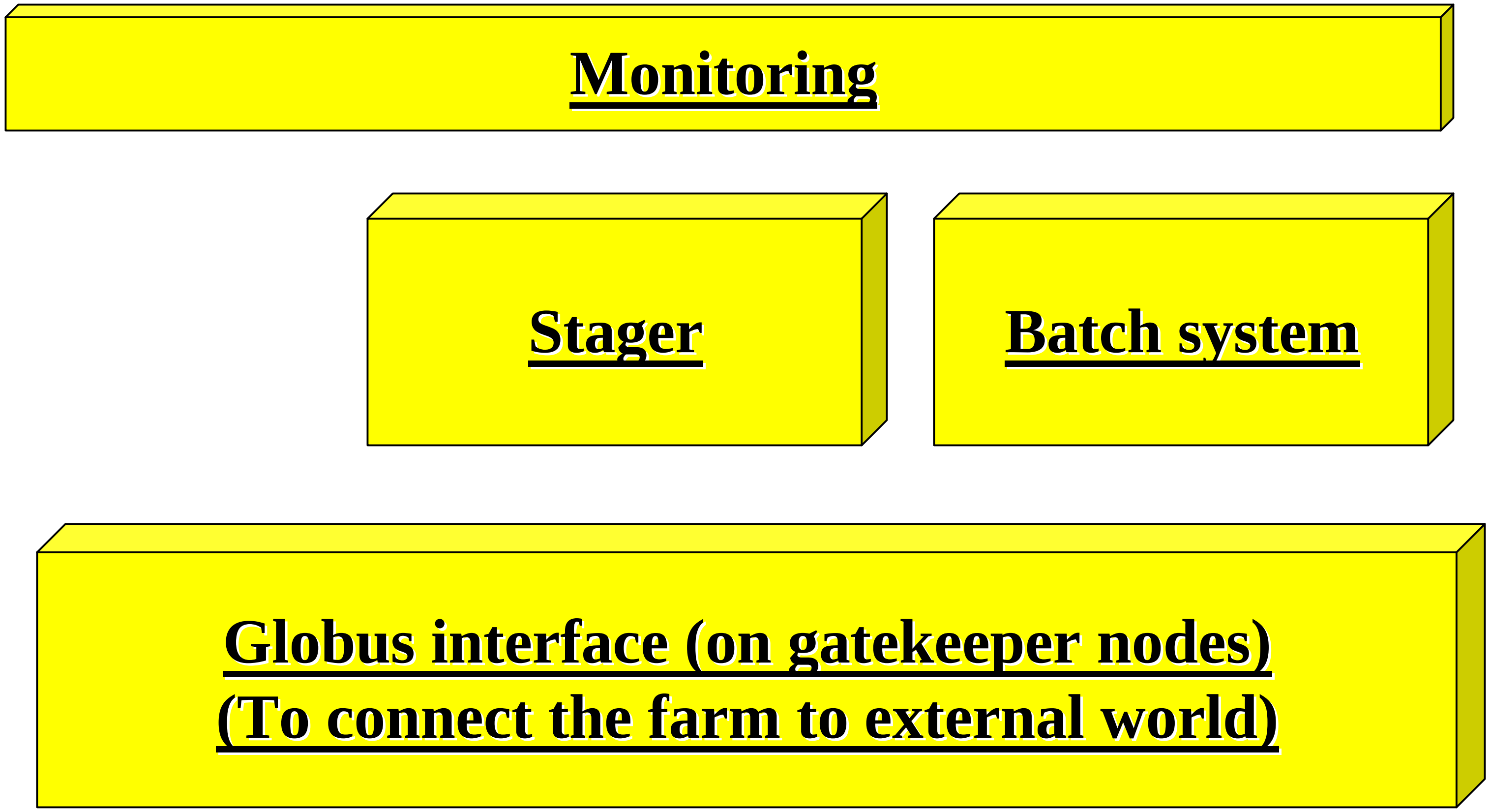
CRS upgrade in nutshell:

- We propose to split the current functions of CRS: staging data and running jobs into 2 separate packages
- Staging data will be done by our own software (to be written)
- Running jobs will be done by an off the shelf batch system (Most likely condor)

What did we gain by that?

- The farm becomes more easily GRID-able (*I hate this word but I know no better*)
- In the next step we will deploy Globus on gatekeeper machines. Running Globus and Condor will enable us to accept (and send) job requests to/from remote farms
- We have moved one step closer towards building a real life example of HEP Grid

THE CRS ARCHTECTURE


Monitoring
Stager
Batch system
Globus interface (on gatekeeper nodes)
(To connect the farm to external world)

Proposed changes to stager:

- Currently CRS accepts 2 types of input files:
- HPSS file and UNIX (disk resident) file
- We will add a new one: GRIDFILE (located anywhere in the world)
- The rest of staging will remain unchanged (except that it is going to be rewritten)

Adding new farms

- The batch system to which we will submit jobs does not need to be on local farm!
- GLOBUS/Condor-G make possible job submission to remote farms
- Once we have stager which can stage jobs to remote locations (GRIDFILE) we could also submit jobs to remote farms

NEW CRS and ATLAS

- Atlas Grid testbed = Condor+CondorG+Globus
- Atlas does not need HPSS very much
- Currently ATLAS people show no interest in present CRS farms, but this will change once CRS becomes a Condor based product

CRS and Atlas

- **Once the BNL farms run Condor and Globus, they can accept jobs from Atlas testbed**
- **Our work then serves not only RHIC community, but for free we have done something useful for Atlas as well.**

Proposed changes to job description:

- I have collected some minor requests from experiments
- Chaining of jobs (requested by experiments)
- Multiple input/output
- ????

Proposed changes to user interface:

- **Guiding principle:** *“Everything must change so that everything can remain the same”*
- **I want as little changes as possible, unless needed or requested by users.**
- **Details need to be discussed while designing the system.**

Timescale

- Draft design: Jan-Feb 2003
- First prototype on small farm: June 2003
- Working system: Dec 2003
- Old CRS phased out – 1st half of 2004